AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A method for generating a solution to a problem having objects scheduled in original itineraries, each said original itinerary having at least an origin and a destination, the method comprising:

receiving a disruption specification based upon an event, the disruption specification including data identifying the objects to be rescheduled;

receiving a request for rescheduling of the objects from a user;

grouping the objects in the disruption specification to be rescheduled into subproblems, wherein each said subproblem is defined by each one of the objects therein having the same original origin and destination;

applying a first algorithm to each said subproblem without allowing varying the origin and destination of the objects in the subproblem, and reaching a plurality of initial solutions that represent a rescheduling for each said object in each said subproblem;

identifying a subclass of objects that [[are]] have been unsuitably rescheduled in the initial solutions; and

applying a second algorithm for rescheduling the subclass of objects that allows varying the original itinerary to generate rescheduling solutions for the subclass of objects.

2. (Currently Amended) A method as recited in claim 1, further comprising the step of applying a [[the]] third algorithm to an Internet Protocol (IP) problem based upon all of the objects in the disruption specification.

3. (Original) A method as recited in claim 2, wherein the third algorithm is an IP algorithm with a branch and bound technique.

4. (Original) A method as recited in claim 2, further comprising the steps of excluding the subclass of objects from the objects that need to be rescheduled in the disruption specification and applying a fourth algorithm to the remaining objects in the reduced disruption specification to determine rescheduling solutions for the remaining objects.

5. (Original) A method according to claim 4, wherein the first and fourth algorithms are transportation simplex algorithms.

6. (Original) A method as recited in claim 1, wherein the subclass of objects to be rerouted are identified based upon a suitably of rescheduling criteria.

7. (Original) A method as recited in claim 6, wherein identifying the subclass includes determining a cost for each rescheduled object and comparing the cost to a threshold.

8. (Original) A method as recited in claim 1, wherein the objects are passengers traveling one or more legs between the origin and the destination.

9. (Original) A method as recited in claim 1, wherein the rescheduling solutions include upgrading, downgrading, delaying, and offloading the objects.

10. (Original) A method according to claim 1, wherein the second algorithm is selected from the group consisting of the Dijkstra algorithm and a K-shortest path algorithm.

11. - 31. (Cancelled)

32. (Currently Amended) A method for generating a solution to a problem having objects scheduled in original itineraries, each said original itinerary having at least an origin and a destination, the method comprising;

receiving a disruption specification based upon an event, the disruption specification including data identifying the objects to be rescheduled;

receiving a request for rescheduling of the objects from a user;

grouping the objects in the disruption specification to be rescheduled into subproblems, wherein each said subproblem is defined by each one of the objects therein having the same original origin and destination;

applying a first algorithm to each subproblem without allowing varying the origin and destination of the objects in the subproblem, and reaching a plurality of initial solutions <u>that represent a rescheduling for each said object in each said subproblem</u>;

identifying a subclass of objects that [[are]] <u>have been</u> unsuitably rescheduled in the initial solutions;

applying a second algorithm for rescheduling the subclass that allows varying the original itinerary to generate rescheduling solutions for the subclass; and

applying a third algorithm to said rescheduling solutions to remove selected ones of said rescheduling solutions, to create a set of optimal rescheduling solutions.